ESE 559: Special Topics in Systems and Control: Learning and Planning in Robotics

Project 1: Kinodynamic Path Planning

February 11, 2025

Project Submission Requirements

Submit a ZIP file that includes (i) **a single PDF file** describing your approach and (ii) **your code**. If you use **Colab or Jupyter Notebook**, include your code and explanations within the notebook and export it as a PDF. Your report and code is due on **March 7**. If you are working in a team of two, please provide a description of how the workload was split and what tasks each member accomplished.

1 Problem Formulation

1.1 System Dynamics

We consider a continuous time deterministic system with state $\mathbf{x}(t) \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ and input $\mathbf{u}(t) \in \mathbf{U} \subseteq \mathbb{R}^{n_u}$ at time $t \geq 0$ with a dynamics defined as follows:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t), \tag{1}$$

where $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ and $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times n_u}$.

In this project, we consider a car-like system where the state is given by $\mathbf{x} = \begin{bmatrix} x & y & \theta & v & \kappa \end{bmatrix}^\top$, where x and y represent the planar position of the robot in meters, θ is the orientation angle in radians, v is the speed in m/s, and κ is the curvature of the path in m^{-1}. The control inputs are $\mathbf{u} = \begin{bmatrix} u_v & u_\kappa \end{bmatrix}^\top$, where u_v and u_κ is the derivative of speed and curvature, respectively. The nonlinear dynamics of the robot are described by:

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = v\kappa, \quad \dot{v} = u_v, \quad \dot{\kappa} = u_\kappa.$$

The linearized dynamics are given by (1) with

$$f(\mathbf{x}) = A = \begin{bmatrix} 0 & 0 & -v\sin\theta & \cos\theta & 0 \\ 0 & 0 & v\cos\theta & \sin\theta & 0 \\ 0 & 0 & 0 & \kappa & v \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad g(\mathbf{x}) = B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The state-space $\mathcal{X}$ is constructed as follows:

- $x \in [0, 10]$
- $y \in [0, 10]$
- $\theta \in [-\pi, \pi]$
- $v \in [0, 2]$
- $\kappa \in [-0.5, 0.5]$

1.2 Task Specification

The initial state $\mathbf{x}_0$ of the robot is $x = y = 1$, $\theta = 0, v = 0, \kappa = 0$. The robot is tasked with eventually reaching a state $\mathbf{x}_g$ that satisfies $x = y = 9$ (there are no desired values for θ, v, κ) while always avoiding the known obstacles.

1.3 Problem 1

Problem 1 *Develop a kinodynamic planner to generate an obstacle-free and dynamically feasible path $\rho : [0, 1] \to \mathcal{X}$, such that $\rho(0) = \mathbf{x}_0$, $\rho(1) = \mathbf{x}_g$ along with the control policy $\pi : [0, 1] \to \mathcal{U}$ mapping each state $\mathbf{x}$ of the path ρ to the control input that needs to be applied there.*

1.4 Problem 2

Problem 2 *Problem 1 does not consider any optimality metrics for the designed path and control policy. In this problem, the goal is to design paths that minimize the total control effort required to follow ρ using π, defined as:*

$$c(\mathcal{T}) := \int_0^T \mathbf{u}_t^T R \mathbf{u}_t dt, \tag{2}$$

In (2), T models the time when the robot reaches $\mathbf{x}_g$ and $R := \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix}$.

In your submitted report, (i) describe your algorithm (Problem 1 & 2); (ii) provide figures illustrating paths designed by your algorithm (Problem 1 & 2); and (iii) provide the cost of the designed paths according to (2) (Problem 1 & 2). Also, comment on the completeness and optimality properties of your algorithm. In your evaluation, you should consider the environments defined in Section 1.5.

1.5 Environments

Consider a $10\text{m} \times 10\text{m}$ environment populated with M known circular obstacles defined as follows.

1.5.1 Environment 1

- **Obstacles:**
  - Center: (5, 5), Radius: 0.7m
  - Center: (7, 2), Radius: 0.5m
  - Center: (3, 8), Radius: 0.6m

1.5.2 Environment 2

- **Obstacles:**
  - Center: (3, 3), Radius: 0.7m
  - Center: (6, 6), Radius: 0.9m
  - Center: (8, 2), Radius: 0.6m
  - Center: (2, 7), Radius: 0.5m
  - Center: (4, 9), Radius: 0.5m

1.6 Environment 3

- **Obstacles:**
  - Center: (4, 4), Radius: 0.6m
  - Center: (6, 6), Radius: 0.6m
  - Center: (7, 3), Radius: 0.7m
  - Center: (2, 7), Radius: 0.5m
  - Center: (8, 8), Radius: 0.6m

1.7 Environment 4

- **Obstacles:**
  - Center: (2, 2), Radius: 0.9m
  - Center: (4, 4), Radius: 0.9m
  - Center: (6, 6), Radius: 1.0m
  - Center: (8, 8), Radius: 0.8m
  - Center: (5, 2), Radius: 0.8m
  - Center: (7, 3), Radius: 0.9m